Distributed Message Service for Kafka

FAQs

Issue	01
Date	2025-01-02

Contents

1 Instances

1.1 Why Can't I Select Two AZs?

To improve the reliability of a Kafka instance, you are advised to select three AZs or more when creating the instance. You cannot select two AZs.

Each Kafka instance contains three ZooKeeper nodes. The ZooKeeper cluster manages Kafka instance configurations. If the ZooKeeper cluster is faulty, the Kafka instance cannot run properly. At least two ZooKeepers are required for the cluster to run properly.

Assume that you select only two AZs. AZ 1 has one ZooKeeper node, and AZ 2 has two. If AZ 1 is faulty, the instance can be used properly. If AZ 2 is faulty, the cluster cannot be used. In this case, the availability rate of the Kafka instance is just 50%. Therefore, do not select 2 AZs.

1.2 Why Can't I View the Subnet and Security Group Information When Creating a DMS Instance?

This may be because you do not have the **Server Administrator** and **VPC Administrator** permissions. For details about how to add permissions to a user group, see **Modifying User Group Permissions**.

1.3 How Do I Select Storage Space for a Kafka Instance?

The storage space is the space for storing messages (including messages in replicas), logs and metadata. To select a storage space, specify the disk type and disk size. For more information, see .

For example, if the required disk size to store data for the retention period is 100 GB, the disk capacity must be at least: **100 GB x Number of replicas + 100 GB (reserved space)**. In a Kafka cluster, each node uses a 33 GB disk to store logs and ZooKeeper data. Therefore, the actual available storage space is less than the purchased storage space.

The number of replicas (3 by default) can be configured when you create a topic. If automatic topic creation has been enabled, each automatically created topic has three replicas by default. You can change this quantity by setting **default.replication.factor** on the **Parameters** tab page.

1.4 How Do I Choose Between High I/O and Ultra-high I/O?

- High I/O: The average latency is 1 to 3 ms, and the maximum bandwidth is 150 MB/s (read + write).

- Ultra-high I/O: The average latency is 1 ms, and the maximum bandwidth is 350 MB/s (read + write).

You are advised to select ultra-high I/O, because ultra-high I/O disks deliver much higher bandwidth than high I/O.

1.5 Which Capacity Threshold Policy Should I Use?

The following policies are supported:

- Stop production

  When the memory usage reaches the disk capacity threshold (95%), new messages will no longer be created, but existing messages can still be retrieved until they are discarded. The default retention time is three days. This policy is suitable for scenarios where no data losses can be tolerated.

- Automatically delete

  When the memory usage reaches the disk capacity threshold (95%), messages can be produced and consumed, but 10% of the earliest messages will be deleted to ensure sufficient disk space. This policy is suitable for scenarios where no service interruption can be tolerated. Data may be lost.

Select a proper policy as required for data and service reliability. Both policies are only used for handling extreme scenarios. **To avoid extreme scenarios, buy sufficient disk space in the first place.**

1.6 Which Kafka Versions Are Supported?

Kafka v1.1.0, v2.7, and v3.x.

For details about how to create a Kafka instance, see **Buying an Instance**.

1.7 What Is the ZooKeeper Address of a Kafka Instance?

Kafka instances are managed using ZooKeeper. Opening ZooKeeper may cause misoperations and service losses. ZooKeeper is used only within Kafka clusters and does not provide services externally.

You can use open-source Kafka clients to connect to Kafka instances and call the native APIs to produce and consume messages.

1.8 Are Kafka Instances in Cluster Mode?

Kafka instances are classified into single-node and cluster types. A single-node instance has only one broker in single-node mode. A cluster instance consists of three or more brokers in cluster mode.

1.9 Can I Modify the Port for Accessing a Kafka Instance?

No. You must access a Kafka instance through one of the following ports:

- Accessing a Kafka instance **without** SASL:

  The port varies with the access mode:
  - Intra-VPC access using IPv4 addresses: port **9092**
  - Intra-VPC access using IPv6 addresses: port **9192**
  - Public access: port **9094**
  - Cross-VPC access using a VPC endpoint: port **9011**
  - Cross-VPC access using a peering connection: port: **9092**.
  - DNAT access: port **9011**
- Accessing a Kafka instance **with** SASL:

  The port varies with the access mode:
  - Intra-VPC access using IPv4 addresses: port **9093**
  - Intra-VPC access using IPv6 addresses: port **9193**
  - Public access: port **9095**
  - Cross-VPC access using a VPC endpoint: port **9011**
  - Cross-VPC access using a peering connection: port: **9093**.
  - DNAT access: port **9011**

Ensure that proper rules have been configured for the security group of the instance. For details, see **How Do I Select and Configure a Security Group?**

1.10 How Long Are Kafka SSL Certificates Valid for?

The certificates are valid for more than 15 years. You do not need to worry about certificate expiration. The certificates are used for one-way authentication when enabling SASL_SSL for Kafka instances.

To check the validity of the SSL certificate, perform the following steps:

Step 1 Decompress the package downloaded from the Kafka instance console to obtain **phy_ca.crt**.

Step 2 Double-click **phy_ca.crt**. The **Certificate** dialog box is displayed.

Step 3 On the **General** tab page, view the certificate validity period.

Figure 1-1 Certificate validity period
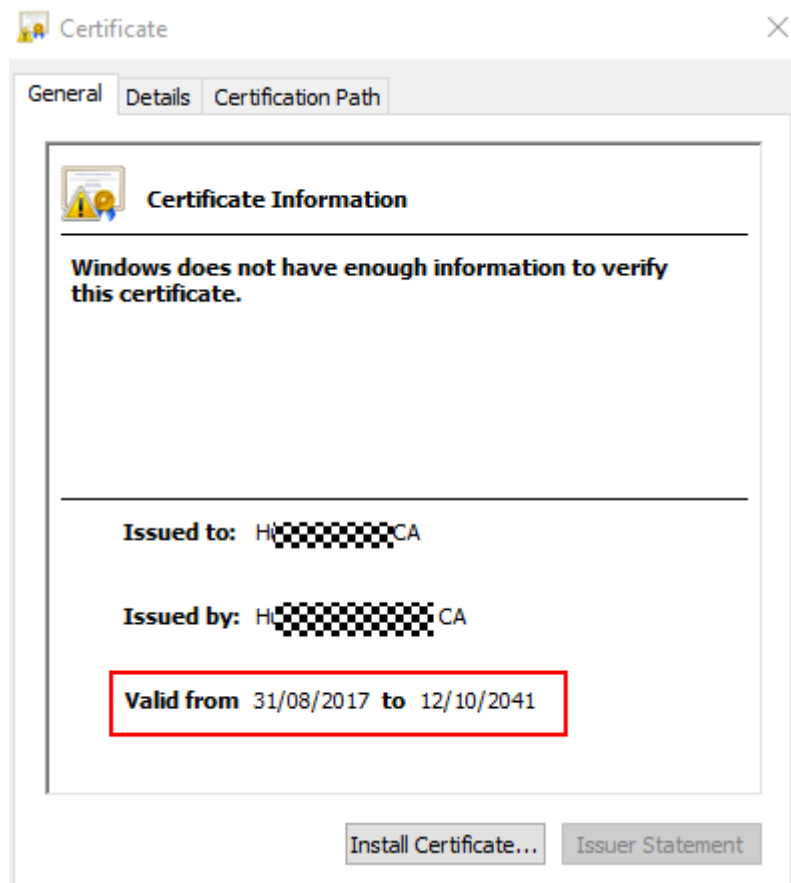
----**End**

1.11 How Do I Synchronize Data from One Kafka Instance to Another?

- If your Kafka instances support data synchronization, enable Smart Connect and then create a Smart Connect task for replicating Kafka data. For details, see **Creating a Smart Connect Task (Kafka)**.

- If your Kafka instances do not support data synchronization, producers can send messages to both Kafka instances. Messages can be consumed from the source instance until all messages in the source instance have been consumed or aged. Then migrate your service to the new Kafka instance.

1.12 How Do I Change the SASL_SSL Setting of a Kafka Instance?

- The SASL_SSL setting of a cluster Kafka instance can be changed after instance creation. For details, see **Changing the Access Mode of an Instance**.

- A single-node instance does not support SASL_SSL.
- SASL_SSL cannot be modified for instances with IPv6 enabled.

1.13 How Do I Modify the SASL Mechanism?

There are two SASL authentication mechanisms for Kafka instances: SCRAM-SHA-512 and PLAIN.

The mechanism cannot be modified after ciphertext access is enabled. If you want to change it, create an instance again.

1.14 How Do I Change the Security Protocol?

There are two security protocols for Kafka instances: SASL_SSL and SASL_PLAINTEXT.

- For instances with IPv6 disabled, the security protocol can be changed on the console. In the **Connection** area on the Kafka instance details page, disable **Ciphertext Access** and then configure security protocol when you enable **Ciphertext Access** again.
- For instances with IPv6 enabled, the security protocol cannot be modified after the instances are created.

1.15 Will a Kafka Instance Be Restarted After Its Enterprise Project Is Modified?

No. A Kafka instance will not be restarted if you modify its enterprise project.

1.16 Are Kafka Brokers and ZooKeeper Deployed on the Same VM or on Different VMs?

Kafka brokers and ZooKeeper are deployed on the same VM.

1.17 Can I Delete a Kafka Instance Billed in Yearly/Monthly Mode?

Yes. Log in to the Kafka console, locate the row that contains the instance, and choose **More** > **Unsubscribe**.

1.18 Which Cipher Suites Are Supported by Kafka?

For security purposes, **TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256**, **TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256**, and **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256** are supported for instances created on and before March 20, 2021.

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 is also supported for instances created after March 20, 2021.

1.19 Can I Change Single-AZ Deployment to Multi-AZ Deployment for an Instance?

No. The AZ cannot be changed once the instance is purchased. To use multiple AZs, purchase another instance.

1.20 Does DMS for Kafka Support Cross-AZ Disaster Recovery? Where Can I Check Whether an Existing Instance is Across-AZs?

DMS for Kafka supports cross-AZ disaster recovery. If you select multiple AZs when buying an instance, cross-AZ disaster recovery will be available.

You can view the AZs in the **Network** section on the **Basic Information** page of the instance. If there are multiple AZs, cross-AZ disaster recovery is available.

Figure 1-2 Instance basic information

1.21 Do Kafka Instances Support Disk Encryption?

No. Kafka instances do not support disk encryption.

1.22 Can I Change the VPC and Subnet After a Kafka Instance Is Created?

No. Once an instance is created, its VPC and subnet cannot be changed.

1.23 Where Can I Find Kafka Streams Use Cases?

You can find Kafka Streams use cases on the **official Kafka website**.

1.24 Can I Upgrade Kafka Instances?

No. Kafka instances cannot be upgraded once they are created. To use a higher Kafka version, create another Kafka instance.

1.25 How Do I Bind an EIP Again?

On the DMS for Kafka console, click the name of the target Kafka instance. Disable **Public Access** in the **Connection** section on the **Basic Information** tab page, and then enable it again. Select the EIP to be bound.

2 Specification Modification

2.1 Does Specification Modification Affect Services?

Table 2-1 describes the impact of specification modification. It takes 5 to 10 minutes to modify specifications on one broker. The more brokers, the longer time the modification takes.

Table 2-1 Impact of specification modification

Modified Object	Impact
Bandwidth or broker quantity	● Increasing the bandwidth or adding brokers does not affect the original brokers or services. ● When you increase the bandwidth or change the broker quantity, the storage space is proportionally expanded based on the current disk space. For example, assume that the original number of brokers of an instance is 3 and the disk size of each broker is 200 GB. If the broker quantity changes to 10 and the disk size of each broker is still 200 GB, the total disk size becomes 2,000 GB. ● New topics are created on new brokers, and the original topics are still on the original brokers, resulting in unbalanced partitions. You can **reassign partitions** to migrate the replicas of the original topic partitions to the new brokers.
Storage space	● You can expand the storage space 20 times. ● Storage space expansion does not affect services.

Modified Object	Impact

Broker flavor	● Single-replica topics do not support message creation and retrieval during this period. Services will be interrupted.
	
	● If a topic has multiple replicas, modifying the broker flavor does not interrupt services, but may cause disorder of partition messages. Evaluate this impact and avoid peak hours.
	
	● Broker rolling restarts will cause partition leader changes, interrupting connections for less than a minute when the network is stable. For multi-replica topics, configure the retry mechanism on the producer client. To do so:
	  – If you use an open-source Kafka client, configure the **retries** parameter to a value in the range from 3 to 5.
	  – If you use Flink, configure the retry policy by referring to the following code:
	`StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();`
	`env.setRestartStrategy(RestartStrategies.fixedDelayRestart(3, Time.seconds(20)));`
	
	● If the total number of partitions created for an instance is greater than the upper limit allowed by a new flavor, scale-down cannot be performed. The maximum number of partitions varies with instance specifications. For details, see **Specifications**.
	For example, if 800 partitions have been created for a **kafka.4u8g.cluster*3** instance, you can no longer scale down the instance to **kafka.2u4g.cluster*3** because this flavor allows only 750 partitions.

2.2 Will Data Migration Be Involved When I Increase Specifications?

No. Data will not be migrated when you increase specifications.

2.3 Why Does Message Production Fail During Scaling?

Possible cause: When you increase or decrease the broker flavor, a rolling restart is performed on brokers. During the restart, partition leaders are changed. The producer has cached the old partition leader IDs, so messages are still sent to old partition leaders. As a result, messages fail to be produced.

Solution: Configure the retry mechanism on the producer by setting **retries** to **Integer.MAX_VALUE**.

2.4 What Can I Do When I Fail to Increase Specifications Due to Insufficient Resources?

Symptom: Specifications fail to be increased, and a message is displayed indicating that the underlying ECS/EVS resources are insufficient. However, the required ECSs can be purchased on the ECS console.

Possible cause: The underlying resource quota is different from the available flavor quota displayed on the console.

Solution: Contact customer service to increase the quota.

3 Connections

3.1 How Do I Select and Configure a Security Group?

Kafka instances can be accessed within a VPC, across VPCs, through DNAT, or over public networks. Before accessing a Kafka instance, configure a security group.

Intra-VPC Access

Step 1 Check whether the client and instance use the same security group.

- If they use the same security group, check whether the security group has the default inbound rule that allows communication among ECSs within the security group and the default outbound rule that allows all outbound traffic. If these rules are available, you do not need to add more rules. If these rules are not available, add rules according to **Table 3-1**.

Table 3-1 Security group rules

Direction	Protocol	Type	Port	Source	Description
Inbound	TCP	IPv4	9092	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a private network within a VPC (in plaintext)
Inbound	TCP	IPv6	9192	IP address or IP address group of the Kafka client	Accessing a Kafka instance within a VPC (with SSL encryption disabled)

Directi on	Proto col	Type	Port	Source	Description
Inboun d	TCP	IPv4	9093	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a private network within a VPC (in ciphertext)
Inboun d	TCP	IPv6	9193	IP address or IP address group of the Kafka client	Accessing a Kafka instance within a VPC (with SSL encryption enabled)

- If they use different security groups, go to **Step 2**.

Step 2 Configure security group rules as follows.

Assume that the security groups of the client and Kafka instance are **sg-53d4** and **Default_All**, respectively. You can specify a security group or IP address as the destination in the following rule. A security group is used as an example.

To ensure that your client can access the Kafka instance, add the following rule to the security group configured for the client:

Table 3-2 Security group rule

Direction	Action	Protocol & Port	Destination
Outbound	Allow	All	Default_All

Figure 3-1 Configuring a security group for the client
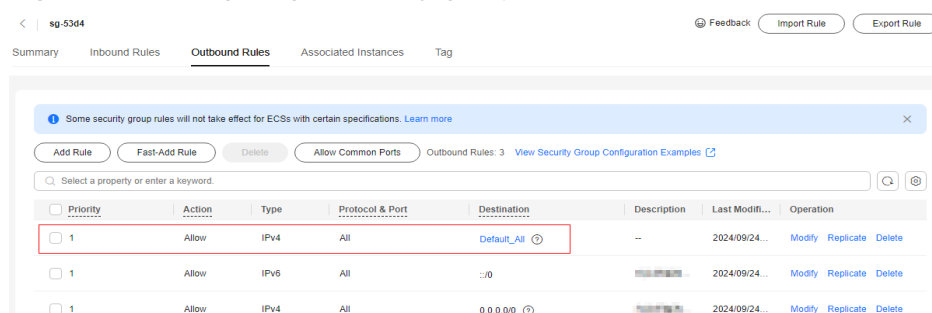
To ensure that your client can access the Kafka instance, add the following rule to the security group configured for the instance.

Table 3-3 Security group rule

Direction	Action	Protocol & Port	Source
Inbound	Allow	All	sg-53d4

Figure 3-2 Configuring the security group for the Kafka instance
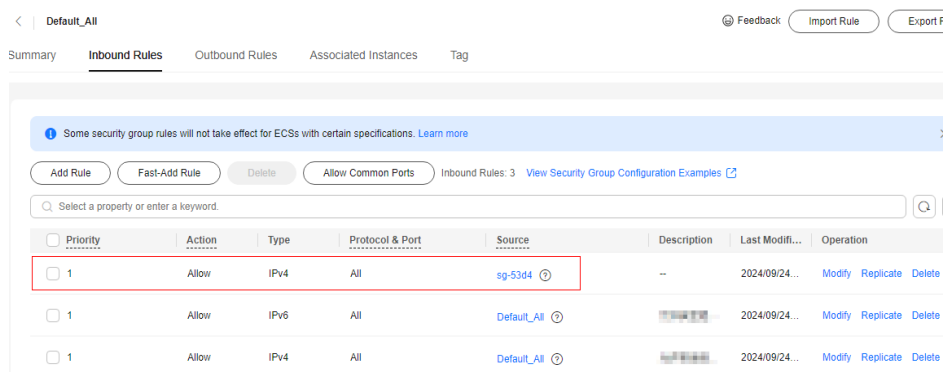
----**End**

Cross-VPC and DNAT-based Instance Access

Configure security group rules according to **Table 3-4**.

Table 3-4 Security group rules

Direction	Protocol	Port	Source	Description
Inbound	TCP	9011	198.19.128.0/17	Accessing a Kafka instance using a VPC endpoint across VPCs (in cipher- or plaintext)
Inbound	TCP	9011	IP address or IP address group of the Kafka client	Accessing a Kafka instance using DNAT (in cipher- or plaintext)
Inbound	TCP	9092	IP address or IP address group of the Kafka client	Accessing a Kafka instance using a peering connection across VPCs (in plaintext)
Inbound	TCP	9093	IP address or IP address group of the Kafka client	Accessing a Kafka instance using a peering connection across VPCs (in ciphertext)

Public Access

Configure security group rules according to **Table 3-5**.

Table 3-5 Security group rules

Direction	Protocol	Type	Port	Source	Description
Inbound	TCP	IPv4	9094	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a public network (in plaintext)
Inbound	TCP	IPv4	9095	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a public network (in ciphertext)
Inbound	TCP	IPv6	9192	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a public network (without SSL)
Inbound	TCP	IPv6	9193	IP address or IP address group of the Kafka client	Accessing a Kafka instance over a public network (with SSL)

3.2 Can I Access a Kafka Instance Over a Public Network?

Yes. For details, see the **instance access instructions**.

3.3 How Many Connection Addresses Does a Kafka Instance Have by Default?

The number of connection addresses of a Kafka instance is the same as the number of brokers of the instance. The following table lists the number of brokers corresponding to each flavor.

Table 3-6 Cluster Kafka instance specifications

Flavor	Brokers	Maximum TPS per Broker	Maximum Partitions per Broker	Recommended Consumer Groups per Broker	Maximum Client Connections per Broker	Storage Space (GB)	Traffic per Broker (MB/s)
kafka.2u4g.cluster.small	3–30	20,000	100	15	2,000	300–300,000	40
kafka.2u4g.cluster	3–30	30,000	250	20	2,000	300–300,000	100
kafka.4u8g.cluster	3–30	100,000	500	100	4,000	300–600,000	200
kafka.8u16g.cluster	3–50	150,000	1000	150	4,000	300–1,500,000	375
kafka.12u24g.cluster	3–50	200,000	1,500	200	4,000	300–1,500,000	625
kafka.16u32g.cluster	3–50	250,000	2,000	200	4,000	300–1,500,000	750

Table 3-7 Single-node Kafka instance specifications

Flavor	Brokers	Max. TPS per Broker	Max. Partitions per Broker	Recommended Consumer Groups per Broker	Max. Client Connections per Broker	Storage Space (GB)	Traffic Limit per Broker (MB/s)
kafka.2u4g.single.small	1	20,000	100	15	2,000	100–10,000	40

Flavor	Brokers	Max. TPS per Broker	Max. Partitions per Broker	Recommended Consumer Groups per Broker	Max. Client Connections per Broker	Storage Space (GB)	Traffic Limit per Broker (MB/s)
kafka.2u4g.single	1	30,000	250	20	2,000	100–10,000	100

3.4 Do Kafka Instances Support Cross-Region Access?

Yes. You can access a Kafka instance across regions over a public network or by using direct connections.

3.5 Do Kafka Instances Support Cross-VPC Access?

Yes. You can use one of the following methods to access a Kafka instance across VPCs:

- Establish a VPC peering connection to allow two VPCs to communicate with each other. For details, see **VPC Peering Connection**.
- Use VPC Endpoint (VPCEP) to establish a cross-VPC connection. For details, see **Accessing Kafka Using a VPC Endpoint Across VPCs**.

3.6 Do Kafka Instances Support Cross-Subnet Access?

Yes.

If the client and the instance are in the same VPC, cross-subnet access is supported. By default, subnets in the same VPC can communicate with each other.

3.7 Does DMS for Kafka Support Authentication with Kerberos?

No, Kerberos authentication is not supported. Kafka supports client authentication with SASL and API calling authentication using tokens and AK/SK.

To access an instance in SASL mode, you need the certificates provided by DMS for Kafka. For details, see **Connecting to Kafka Using the Client (SASL Enabled)**.

3.8 Does DMS for Kafka Support Password-Free Access?

Yes. No password is required when accessing a Kafka instance in plaintext. For details, see **Connecting to Kafka Using the Client (SASL Disabled)**.

3.9 How Do I Obtain the Public Access Address After Public Access Is Enabled?

Click the name of your Kafka instance. In the **Connection** section on the **Basic Information** tab page, view **Instance Address (Public Network)**.

For details about how to connect to a Kafka instance, see **the instance access instructions**.

3.10 Does DMS for Kafka Support Authentication on Clients by the Server?

No.

3.11 Can I Use PEM SSL Truststore When Connecting to a Kafka Instance with SASL_SSL Enabled?

Yes. For details, see **Obtaining and Using a PEM SSL Certificate**.

3.12 What Are the Differences Between JKS and CRT Certificates?

JKS certificates are used for connecting to instances in Java and CRT certificates are used for connecting to instances in Python.

3.13 Which TLS Version Does DMS for Kafka Support?

TLS 1.2.

3.14 Is There a Limit on the Number of Client Connections to a Kafka Instance?

Yes. The maximum allowed number of client connections varies by instance specifications.

Table 3-8 Number of connections of earlier Kafka instances

Assured Bandwidth	Max. Connections
100 MB/s	3,000
300 MB/s	10,000
600 MB/s	20,000
1,200 MB/s	20,000

Table 3-9 Number of connections of later Kafka instances

Flavor	Max. Client Connections per Broker
kafka.2u4g.cluster.small	2,000
kafka.2u4g.single.small	2,000
kafka.2u4g.cluster	2,000
kafka.2u4g.single	2,000
kafka.4u8g.cluster	4,000
kafka.8u16g.cluster	4,000
kafka.12u24g.cluster	4,000
kafka.16u32g.cluster	4,000

3.15 How Many Connections Are Allowed from Each IP Address?

For instances purchased in July 2020 and later, each Kafka broker allows a maximum of 1000 connections from each IP address by default. For instances purchased before July 2020, each Kafka broker allows a maximum of 200 connections from each IP address by default. Excess connections will be rejected. You can change the limit by **modifying the Kafka parameters**.

3.16 Can I Change the Private Network Addresses of a Kafka Instance?

After an instance is created, its private network connection addresses cannot be modified. You can specify private network connection addresses when creating an instance or adding brokers.

3.17 Is the Same SSL Certificate Used for Different Instances?

Yes. All Kafka instances and users use the same SSL certificate.

To obtain the SSL certificate, perform the following steps:

Step 1 On the Kafka console, click the name of your instance.

Step 2 In the **Connection** area on the **Basic Information** tab page, click **Download** next to **SSL Certificate**.

----End

3.18 Why Is It Not Recommended to Use a Sarama Client for Messaging?

Symptom

If a Sarama client is used to send and receive messages, the following issues may occur:

- Sarama cannot detect partition changes. Adding topic partitions requires client restart to enable consumption.

- Sarama's default **MaxProcessingTime** is 100 ms. When this limit is reached, consumers can no longer consume messages.

- If consumer offsets reset from the oldest (earliest) position, all messages starting from the earliest position may be repeatedly consumed after the client restarts.

- A consumer that subscribes to multiple topics may not be able to consume any message from specific partitions.

Solution

Use **confluent-kafka-go** as the Kafka client library.

For details, see **Table 3-10**.

Table 3-10 Comparing common Go clients

Client	Pros	Cons
confluent-kafka-go	- An official Kafka client by Confluent that supports full Kafka compatibility and all Kafka features - High stability and performance, and low latency based on librdkafka	High compiling complexity because Go compilers need extra resources to configure the imported C++ libraries
Kafka-go	- A simple and lite Kafka client easy for learning and usage - Reduced application size and complexity with limited library and fewer dependencies	- Fewer advanced functions and configurations than confluent-kafka-go - Applicable only to simple scenarios that require low performance and throughput
Sarama	Better asynchronization and higher concurrency (written in the original Go language)	- Many bugs, limited documentation - Deteriorates application performance when processing a large number of messages due to large memory consumption

4 Topics and Partitions

4.1 Is There a Limit on the Number of Topics in a Kafka Instance?

The number of topics is related to the total number of topic partitions and the number of partitions in each topic. There is an upper limit on the aggregate number of partitions of topics. When this limit is reached, no more topics can be created.

The partition limit varies depending on the flavor, as shown in the following table.

Table 4-1 Cluster Kafka instance specifications

Flavor	Brokers	Maximum TPS per Broker	Maximum Partitions per Broker	Recommended Consumer Groups per Broker	Maximum Client Connections per Broker	Storage Space (GB)	Traffic per Broker (MB/s)
kafka.2u4g.cluster.small	3–30	20,000	100	15	2,000	300–300,000	40
kafka.2u4g.cluster	3–30	30,000	250	20	2,000	300–300,000	100
kafka.4u8g.cluster	3–30	100,000	500	100	4,000	300–600,000	200

Flavor	Bro kers	Maximum TPS per Broker	Maximum Partitions per Broker	Recommended Consumer Groups per Broker	Maximum Client Connections per Broker	Storage Space (GB)	Traffic per Broker (MB/s)
kafka.8u 16g.cluster	3–50	150,000	1000	150	4,000	300–1,500,000	375
kafka.12 u24g.cluster	3–50	200,000	1,500	200	4,000	300–1,500,000	625
kafka.16 u32g.cluster	3–50	250,000	2,000	200	4,000	300–1,500,000	750

Table 4-2 Single-node Kafka instance specifications

Flavor	Bro kers	Max. TPS per Broker	Max. Partitions per Broker	Recommended Consumer Groups per Broker	Max. Client Connections per Broker	Storage Space (GB)	Traffic Limit per Broker (MB/s)
kafka.2u 4g.single .small	1	20,000	100	15	2,000	100–10,000	40
kafka.2u 4g.single	1	30,000	250	20	2,000	100–10,000	100

4.2 Why Is Partition Quantity Limited?

Kafka manages messages by partition. If there are too many partitions, message creation, storage, and retrieval will be fragmented, affecting the performance and stability. If the total number of partitions of topics reaches the upper limit, you cannot create more topics.

The partition limit varies depending on the flavor, as shown in the following table.

Table 4-3 Cluster Kafka instance specifications

Flavor	Brokers	Maximum TPS per Broker	Maximum Partitions per Broker	Recommended Consumer Groups per Broker	Maximum Client Connections per Broker	Storage Space (GB)	Traffic per Broker (MB/s)
kafka.2u4g.cluster.small	3–30	20,000	100	15	2,000	300–300,000	40
kafka.2u4g.cluster	3–30	30,000	250	20	2,000	300–300,000	100
kafka.4u8g.cluster	3–30	100,000	500	100	4,000	300–600,000	200
kafka.8u16g.cluster	3–50	150,000	1000	150	4,000	300–1,500,000	375
kafka.12u24g.cluster	3–50	200,000	1,500	200	4,000	300–1,500,000	625
kafka.16u32g.cluster	3–50	250,000	2,000	200	4,000	300–1,500,000	750

Table 4-4 Single-node Kafka instance specifications

Flavor	Brokers	Max. TPS per Broker	Max. Partitions per Broker	Recommended Consumer Groups per Broker	Max. Client Connections per Broker	Storage Space (GB)	Traffic Limit per Broker (MB/s)
kafka.2u4g.single.small	1	20,000	100	15	2,000	100–10,000	40

Flavor	Brokers	Max. TPS per Broker	Max. Partitions per Broker	Recommended Consumer Groups per Broker	Max. Client Connections per Broker	Storage Space (GB)	Traffic Limit per Broker (MB/s)
kafka.2u4g.single	1	30,000	250	20	2,000	100–10,000	100

4.3 Can I Reduce the Partition Quantity?

No. If you want to use fewer partitions, delete the corresponding topic, create another one, and specify the desired number of partitions.

4.4 Why Do I Fail to Create Topics?

Possible cause: The aggregate number of partitions of created topics has reached the upper limit. The maximum number of partitions varies by instance specifications. For details, see **Specifications**.

Solution: Scale up the instance or delete unnecessary topics.

4.5 Do Kafka Instances Support Batch Importing Topics or Automatic Topic Creation?

Automatic topic creation is supported, but batch topic import is not supported. You can only export topics in batches.

Enable automatic topic creation using one of the following methods:

- When creating an instance, enable automatic topic creation.
- After an instance is created, enable automatic topic creation on the **Basic Information** tab page.

4.6 Why Do Deleted Topics Still Exist?

Possible cause: Automatic topic creation has been enabled and a consumer is connecting to the topic. If no existing topics are available for message creation, new topics will be automatically created.

Solution: Disable automatic topic creation.

4.7 Can I View the Disk Space Used by a Topic?

Yes. Use either of the following methods to check the disk space used by a topic:

- In the row containing the desired Kafka instance, click **View Metric** to go to the Cloud Eye console. On the **By Topic** tab page, set **Topic** to the name of the desired topic, and **Monitoring Type** to **Basic monitoring**. The **Message Size** metric reflects the message size of the selected topic.

- Click the desired Kafka instance to go to the instance details page. Choose **Monitoring and Alarms** > **Monitoring Details** in the navigation pane. On the **By Topic** tab page, set **Topic** to the name of the topic whose disk space you want to view and **Monitoring Type** to **Basic monitoring**. The **Message Size** metric reflects the message size of the selected topic.

4.8 Can I Add ACL Permissions for Topics?

If you have enabled SASL_SSL for your Kafka instance, you can configure ACL permissions for your topics. On the **Topics** tab page of the Kafka console, click **Grant User Permission** in the row that contains the topic for which you want to configure user permissions.

For details, see **Configuring Topic Permissions**.

4.9 What Should I Do If Kafka Storage Space Is Used Up Because Retrieved Messages Are Not Deleted?

Messages are not deleted immediately after being retrieved. They are deleted only when the aging time expires.

You can shorten the aging time or expand the storage space.

4.10 How Do I Increase the Partition Quantity?

You can increase the partition quantity by increasing the bandwidth or adding brokers.

To do so, go to the Kafka console, locate the row that contains the desired instance, and choose **More** > **Modify Specifications**. On the page that is displayed, increase the bandwidth or add brokers as required. For details, see **Modifying Kafka Instance Specifications**.

4.11 Will a Kafka Instance Be Restarted After Its Automatic Topic Creation Setting Is Modified?

Enabling or disabling automatic topic creation may cause instance restarts. For details, see the information displayed on the Kafka console.

4.12 What Can I Do If a Consumer Fails to Retrieve Messages from a Topic Due to Insufficient Permissions?

Symptom: Different consumers in a consumer group have different topic permissions. When a consumer attempts to retrieve messages from a topic, the error message "Not authorized to access topics." is displayed, and the message retrieval fails.

```
Tests passed: 1 of 1 test – 40 s 771 ms
"C:\Program Files\H▮▮▮i\jdk1.8.0_242\bin\java.exe" ...
the numbers of topic:0
the numbers of topic:0
the numbers of topic:0
the numbers of topic:0
the numbers of topic:0
org.apache.kafka.common.errors.TopicAuthorizationException: Not authorized to access topics: [huanwei-gongce-basic, topic-huanwei-data]
```

Analysis: When assigning partitions, the leader of the consumer group does not consider the permissions of individual consumers. Instead, the leader assigns partitions based on the overall subscription of the consumer group. In this case, consumers may be assigned topics that they do not have access to.

For example, consumers A, B, and C are in the same consumer group. Consumer A has subscribed to and has permissions to access Topics 0, 1, and 2; consumer B has subscribed to and has permissions to access Topics 3, 4, and 5; consumer C has subscribed to and has permissions to access Topics 6, 7, and 8. Assume that each topic has only one partition. Based on the partition assignment determined by the leader, consumer A may be assigned Topics 0, 3, and 6; consumer B is assigned Topics 1, 4, and 7; and consumer C is assigned Topics 2, 5, and 8. In this case, consumer A does not have permissions to access Topics 3 and 6, resulting in the error.

Figure 4-1 Consumer access permissions
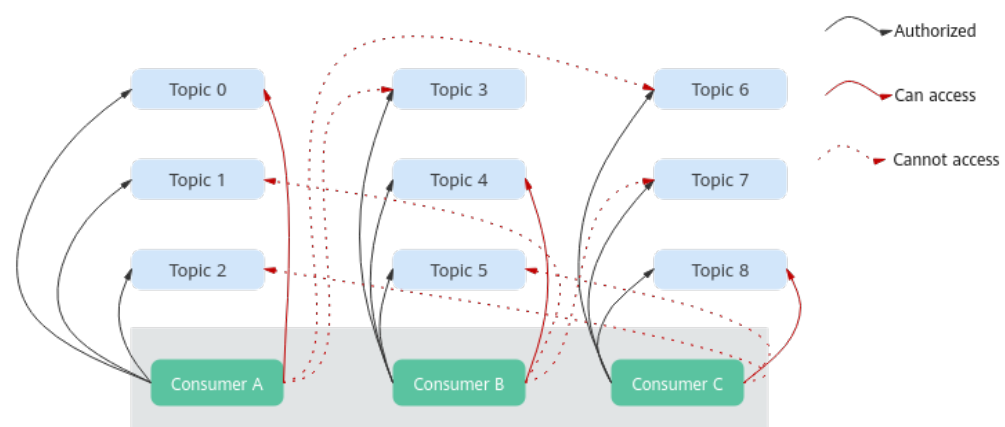
Solution:

- If all consumers must be in the same consumer group (**group.id** is the same), grant the same topic access permissions to all the consumers.

- If the consumers do not need to be in the same consumer group, change the value of **group.id** to ensure that each consumer is in a separate consumer group.

5 Consumer Groups

5.1 Do I Need to Create Consumer Groups, Producers, and Consumers for Kafka Instances?

- When parameter **auto.create.groups.enable** is set to **true**, you do not need to create a consumer group, producer, or consumer because they are generated automatically when you use the instance.

- When parameter **auto.create.groups.enable** is set to **false**, you need to create a consumer group, but do not need to create a producer or consumer.

To change the **auto.create.groups.enable** setting, see **Modifying Kafka Instance Configuration Parameters**.

For details about producing and consuming messages after connecting to a Kafka instance, see **Connecting to Kafka Using the Client (Plaintext Access)**.

5.2 Will a Consumer Group Without Active Consumers Be Automatically Deleted in 14 Days?

This depends on the **offsets.retention.minutes** and **auto.create.groups.enable** parameters.

- For instances created before Apr 25, 2023, **auto.create.groups.enable** is set to **true** by default. **offsets.retention.minutes** determines how long before a consumer group is deleted automatically, which can be changed on the console. For details, see **Modifying Kafka Instance Configuration Parameters**.

- For instances created on or after Apr 25, 2023:
  - If **auto.create.groups.enable** is **false**, you need to manually delete consumer groups.

    If **auto.create.groups.enable** is set to **false** for a Kafka instance created before Feb 5, 2024, and an empty consumer group subscribed to only one topic, deleting the topic also removes the consumer group.

- If **auto.create.groups.enable** is **true**, a consumer group that has never committed an offset will be automatically deleted after 10 minutes.

- If **auto.create.groups.enable** is **true**, and a consumer group that has committed an offset, **offsets.retention.minutes** determines how long before the group is deleted automatically, which can be changed on the console. For details, see **Modifying Kafka Instance Configuration Parameters**.

Kafka uses the **offsets.retention.minutes** parameter to control how long to keep offsets for a consumer group. If offsets are not committed within this period, they will be deleted. If Kafka determines that there are no active consumers in a consumer group (for example, when the consumer group is empty) and there are no offsets, Kafka will delete the consumer group.

5.3 Why Does a Deleted Consumer Group Still Exist?

Possible cause: Automatic consumer group creation has been enabled and your service is connected to the consumer group and consuming messages. Therefore, the consumer group is deleted but will be automatically re-created to consume messages until your service stops.

Solution: Disable automatic consumer group creation. To do so, go to the Kafka console, choose **Parameters** in the left navigation pane, locate the row that contains **auto.create.groups.enable** in the **Dynamic Parameters** area, click **Edit**, set the **Current Value** to **false**, and click **Save**. Then, delete the consumer group. **auto.create.groups.enable** is not displayed for some instances. For details, see the console. In this case, modify your service code to stop connecting to the consumer group. Then, delete the consumer group. To change the **auto.create.groups.enable** parameter setting, see **Modifying Kafka Instance Configuration Parameters**.

5.4 Why Can't I View Consumers When Instance Consumption Is Normal?

Check whether Flink is used for consumption. Flink uses the assign mode and the client assigns specific partitions to be consumed, so you cannot see any consumer on the Kafka console.

5.5 Can I Delete Unnecessary Topics in a Consumer Group?

Just simply unsubscribe from them on the Kafka client.

6 Messages

6.1 What Is the Maximum Size of a Message that Can be Created?

10 MB.

6.2 Why Does Message Poll Often Fail During Rebalancing?

Rebalancing is a process where partitions of topics are re-allocated for a consumer group.

In normal cases, rebalancing occurs inevitably when a consumer is added to or removed from a consumer group. However, if a consumer is regarded as abnormal by the server and removed from the consumer group, message consumption may fail.

This may happen in the following scenarios:

1. Heartbeat requests are not sent in time.

   A consumer sends heartbeat requests to the broker at the interval specified by **heartbeat.interval.ms**. If the broker does not receive any heartbeat request from the consumer within the period specified by **session.timeout.ms**, the broker considers that the consumer is abnormal and removes the consumer from the consumer group, triggering rebalancing.

2. The interval between retrievals is too long.

   The maximum number of messages that a consumer can consume at a time is specified by **max.poll.records**. In most cases, a client processes the consumed data before starting the next consumption. The processing may be prolonged when a large number of messages are consumed at a time and cannot be processed within the time specified by **max.poll.interval.ms**, or when an exception occurs during the process (for example, data needs to be written to the backend database, but the backend database pressure is too high, resulting in slow SQL and high latency). If the consumer does not send

the next consumption request within the time specified by
max.poll.interval.ms, the broker considers that the consumer is inactive and
removes it from the consumer group, triggering rebalancing.

Solutions and Troubleshooting Methods

Scenario 1: Heartbeat requests are not sent in time.

Locating: If the client is in Java, check GC logs to see whether FullGC takes a long
time. If yes, the heartbeat thread may be blocked. As a result, the server fails to
check heartbeats and starts rebalancing.

Solution: Check for client issues. FullGC indicates that memory leakage may have
occurred on the client.

Scenario 2: The interval between retrievals is too long.

Troubleshooting methods:

1. Check the time required for processing a single message and whether the
    time required for processing a specified number (**max.poll.records**) of
    messages exceeds the time specified by **max.poll.interval.ms**.

2. Check whether message processing requires network connections, such as
    writing data to the database and calling backend APIs, and whether the
    downstream consumer system is normal in rebalancing scenarios.

Solution: On the consumer client, decrease the value of **max.poll.records**. If
message processing takes longer, increase the value of **max.poll.interval.ms**.

6.3 Why Can't I Query Messages on the Console?

- **Possible cause 1**: The message has been aged.

  Solution: Change the aging time.

- **Possible cause 2**: The createTime timestamp of the message is incorrect.

  On the console, messages are queried based on the timestamp, which is
  generated by the client. Different clients have different processing policies.
  The default value may be **0** or **-1**. As a result, message may fail to be queried.

  Solution: Check whether the value of createTime is correctly configured.

- **Possible cause 3**: The disk usage exceeds 95%, and **Capacity Threshold
  Policy** is set to **Automatically delete**.

  If **Capacity Threshold Policy** is set to **Automatically delete**, the earliest 10%
  of messages will be deleted when 95% of the disk capacity is used, to ensure
  sufficient disk space. In this case, the messages that do not reach the aging
  time are also deleted and cannot be queried.

  Solution: Modify the capacity threshold policy or expand the disk capacity. If
  Capacity Threshold Policy is set to **Stop production**, new messages will no
  longer be created when the disk usage reaches the capacity threshold (95%),
  but existing messages can still be retrieved until the aging time arrives. This
  policy is suitable for scenarios where no data losses can be tolerated.

6.4 What Can I Do If Kafka Messages Are Accumulated?

Symptom: An alarm is generated for the **Accumulated Messages** metric.

Solution 1:

1. Log in to the Kafka console and click the instance for which the alarm is generated. The instance details page is displayed.

2. Choose **Monitoring and Alarms** > **Monitoring Details** in the navigation pane.

3. On the **By Consumer Group** tab page, view **Consumer Retrieved Messages** to find the consumer group with accumulated messages.

4. In the navigation pane, choose **Consumer Groups**.

5. Check whether there are consumers in the consumer group where messages are accumulated. If yes, contact the service party to accelerate their consumption. If no, contact the customer to delete unused consumer groups.

Solution 2:

1. Log in to the Kafka console and click the instance for which the alarm is generated. The instance details page is displayed.

2. In the left navigation pane, choose **Analysis & Diagnosis** > **Accumulation Diagnosis**.

3. Select the consumer groups and topics to be diagnosed, and click **Pre-check**.

4. After the pre-check is complete, if there are any risks in the **Accumulated Messages** area and the consumer group is not consuming message in the assign mode, click **Start Diagnosis**.

5. After the diagnosis, locate the row that contains the diagnosis result, and click **View Details** in the **Operation** column. The **Diagnosis Details** page is displayed. Check the possible causes, affected partitions or brokers, and suggestions. Follow the suggestions to handle message accumulation.

6.5 Why Do Messages Still Exist After the Retention Period Elapses?

If the aging time has been set for a topic, the value of the **log.retention.hours** parameter does not take effect for the topic. The value of the **log.retention.hours** parameter takes effect only if the aging time has not been set for the topic.

Possible cause 1: Each partition of a topic consists of multiple segment files of the same size (500 MB). When the size of messages stored in a segment file reaches 500 MB, another segment file is created. Kafka deletes segment files instead of messages. Kafka requires that at least one segment file be reserved for storing messages. If the segment file in use contains aged messages, the segment file will not be deleted. Therefore, the aged messages will remain.

Solution: Wait until the segment is no longer in use or delete the topic where messages have reached their retention period.

Possible cause 2: In a topic, there is a message whose **CreateTime** is a future time. For example, assume that it is January 1, and the **CreateTime** is February 1. The message will not be aged after 72 hours from now. As a result, messages created subsequently will also not be aged.

Solution: Delete the topic where the **CreateTime** of a message is a future time.

6.6 Do Kafka Instances Support Delayed Message Delivery?

No.

6.7 How Do I View the Number of Accumulated Messages?

View the number of accumulated messages using any of the following methods:

- On the **Consumer Groups** page of an instance, click the name of the consumer group whose accumulated messages are to be viewed. The consumer group details page is displayed. On the **Consumer Offset** tab page, view the number of messages accumulated in each topic of your target consumer group. For details, see **Viewing Consumer Offsets**.

- On the **Monitoring and Alarms** > **Monitoring Details** tab page of an instance, click the **By Consumer Group** tab. Select the desired consumer group for **Consumer Group** and **All topics** for **Topic**. The **Consumer Available Messages** metric reflects the number of messages accumulated in all topics of this consumer group. For details about viewing the monitoring data, see **Viewing Kafka Metrics**.

- On the **Consumer Groups** tab page of the Cloud Eye console, click the **By Consumer Group** tab. Select the desired consumer group for **Consumer Group** and **All topics** for **Topic**. The **Consumer Available Messages** metric reflects the number of messages accumulated in all topics of this consumer group. For details about viewing the monitoring data, see **Viewing Kafka Metrics**.

- In the **/bin** directory on the **Kafka client**, run command **kafka-consumer-groups.sh --bootstrap-server ${connection-address} --describe --group ${consumer-group-name}** to check number of accumulated messages of each topic in a consumer group. **LAG** indicates the total number of messages accumulated in each topic.

Figure 6-1 Viewing the total number of messages accumulated in each topic

If SASL authentication is enabled for the Kafka instance, the **--command-config {SASL authentication configuration file consumer.properties}** parameter must be added to the preceding commands. For details about the

configuration file **consumer.properties**, see the CLI access instructions provided in **Connecting to Kafka Using the Client (SASL Enabled)**.

6.8 Why Is the Message Creation Time Displayed as Year 1970?

The message creation time is specified by **CreateTime** when a producer creates messages. If this parameter is not set during message creation, the message creation time is year 1970 by default.

6.9 How Do I Modify message.max.bytes?

message.max.bytes can be modified on the **Parameters** page on the console. For details, see **Modifying Kafka Instance Configuration Parameters**.

The maximum value of **message.max.bytes** is 10,485,760 bytes. If a single message is larger than this size, compress the message with algorithms or split it in the service logic.

6.10 Why Are Offsets Not Continuous?

If you have enabled idempotence or transactions on the producer client, and produced messages, message offsets are not continuous on the consumer client or on the **Message Query** page on the Kafka console. This is because enabling idempotence or transactions generates some metadata control messages during message production. These control messages are produced to topics, and are invisible to consumers.

By default, idempotence is enabled for Kafka 3.0 and later producer clients. To disable it, set the parameter **enable.idempotence** to **false**.
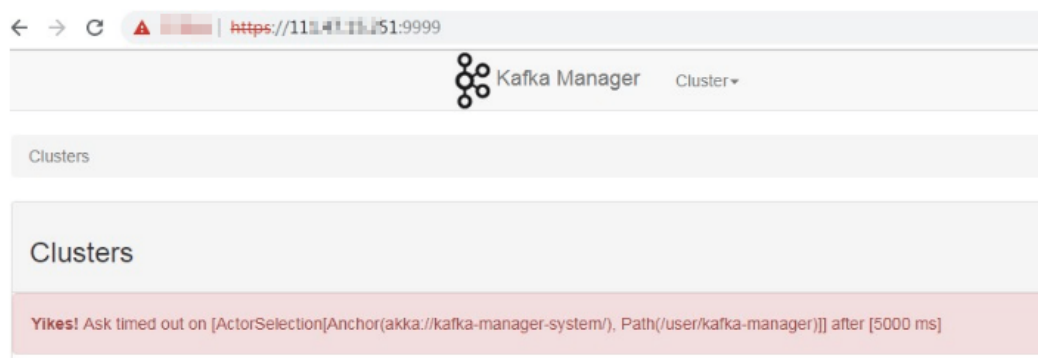
7 Kafka Manager

7.1 Can I Configure a Kafka Manager Account to Be Read-Only?

No. You cannot configure a Kafka Manager account to be read-only.

7.2 Why Can't I See Broker Information After Logging In to Kafka Manager?

Symptom: The Kafka Manager page is displayed, but the broker information cannot be displayed.

Cause: This is an issue with the open-source Kafka. To solve the problem, contact customer service and restart Kafka Manager.

7.3 Yikes! Insufficient partition balance when creating topic : projectman_project_enterprise_project Try again.

Symptom:

The topic cannot be created in Kafka Manager, and the error message "Yikes! Insufficient partition balance when creating topic : projectman_project_enterprise_project Try again." is displayed.

Cause: The number of partitions exceeds the upper limit and no more topics can be created.

Solution: Increase the instance specifications, which will automatically increase the allowed number of partitions.

7.4 Can I Query the Body of a Message by Using Kafka Manager?

No. Kafka Manager does not support message body querying.

7.5 Can I Change the Port of the Kafka Manager Web UI?

No.

7.6 Which Topic Configurations Can Be Modified on Kafka Manager?

On Kafka Manager, the following topic configurations can be modified: **max.message.bytes**, **segment.index.bytes**, **segment.jitter.ms**, **min.cleanable.dirty.ratio**, **retention.bytes**, **file.delete.delay.ms**, **compression.type**, **flush.ms**, **cleanup.policy**, **unclean.leader.election.enable**, **flush.messages**, **retention.ms**, **min.insync.replicas**, **delete.retention.ms**, **preallocate**, **index.interval.bytes**, **segment.bytes**, and **segment.ms**.

Perform the following steps to modify the topic configurations:

1. **Log in to Kafka Manager**.
2. Click **kafka_cluster**.
3. Choose **Topic** > **List**.
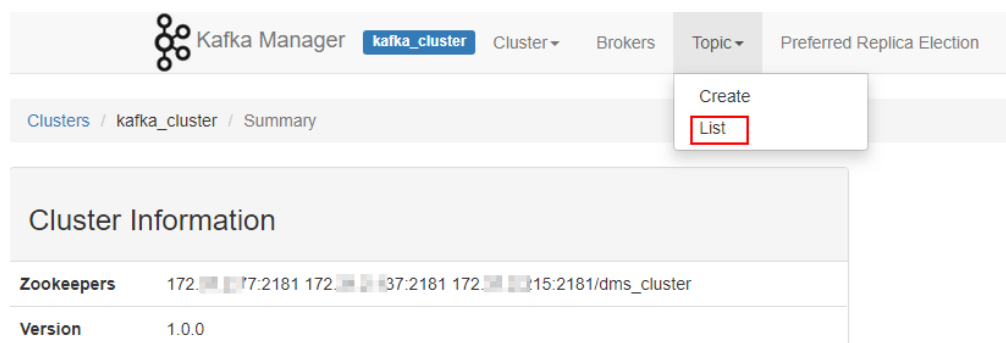
4. Click a topic whose configurations you want to modify.
5. Click **Update Config**.

7.7 Why Is Information Displayed on Kafka Manager Inconsistent with Cloud Eye Monitoring Data?

Symptom: After a consumer group is deleted from the backend, the consumer group is not displayed on Cloud Eye but still exists on Kafka Manager.

Cause: Kafka Manager has cache data.

Solution: Log in to the Kafka console, locate the row that contains the target instance, and choose **More** > **Restart Kafka Manager**.

7.8 How Do I Change a Partition Leader for a Topic in Kafka Manager?

Perform the following steps:

1. **Log in to Kafka Manager**.
2. Choose **Topic** > **List**.

3. Click the topic name (for example, **topic-test**) for which a partition leader is to be modified.

4. Click **Manual Partition Assignments**.

Figure 7-1 Topic details

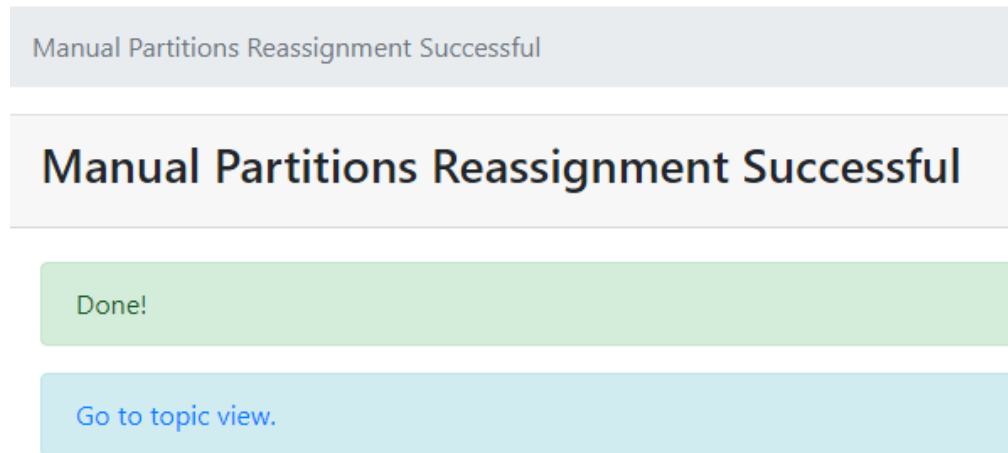**Figure 7-2** Page for modifying partition leaders

For example, in the preceding figure, the leader (Replica 0) of Partition 2 is on Broker 2.

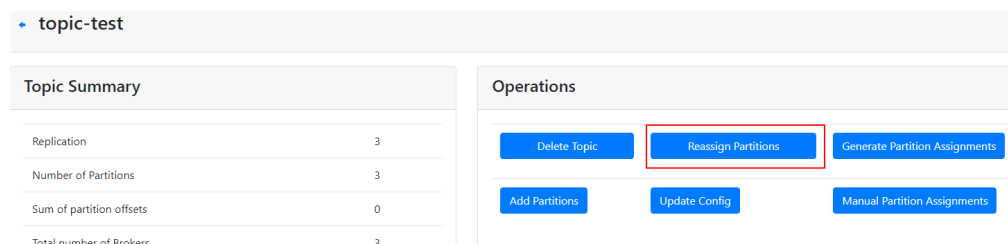5. Change the leader and click **Save Partition Assignment**.

If the modification is successful, the information shown in the following figure is displayed.

Manual Partitions Reassignment Successful

Manual Partitions Reassignment Successful

Done!

Go to topic view.

6. Click **Go to topic view**.
7. Click **Reassign Partitions** to save the change.

◆ topic-test

Topic Summary	
Replication	3
Number of Partitions	3
Sum of partition offsets	0
Total number of Brokers	3

Operations

| Delete Topic | Reassign Partitions | Generate Partition Assignments |
| Add Partitions | Update Config | Manual Partition Assignments |

After the change is saved, the information shown in the following figure is displayed.

Clusters / kafka_cluster / Topics / topic-test / Run Reassign Partitions

Run Reassign Partitions - topic-test

Done!

Go to reassign partitions.
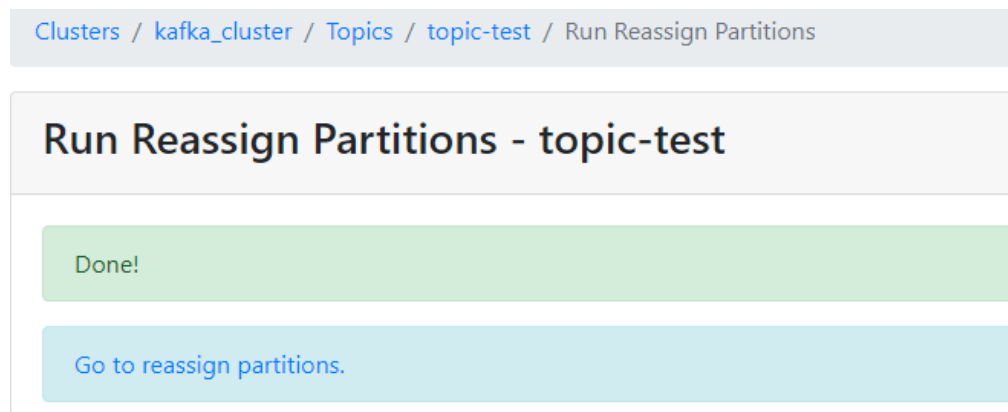
8. In the breadcrumb navigation, click the topic name. On the topic details page that is displayed, view the partition details.

As shown in the preceding figure, the leader of partition 2 has been changed from 2 to 1.

7.9 Why Is the Version on the Console Different from That in Kafka Manager?

The version displayed on the console is used for your instance. Kafka Manager uses the common configuration of open-source Kafka 2.2.0. Therefore, the version displayed in Kafka Manager is 2.2.0, which is irrelevant to the version of your Kafka instance.

7.10 Why Does an Instance Contain Default Topics __trace and __consumer_offsets?

Symptom: Topics named **__trace** and **__consumer_offsets** are found on Kafka Manager.

Handling method: **__trace** and **__consumer_offsets** are preset topics in a Kafka instance. You are not advised to delete them. If they are deleted, the instance may become unavailable.

7.11 Why Do I See a Deleted Consumer Group on Kafka Manager?

After a consumer group is deleted on a client, it no longer exists, but may still be displayed on Kafka Manager because of Kafka Manager's cache.

Use either of the following methods to solve the problem:

- Restart Kafka Manager.
- Kafka Manager displays only the consumer groups that have retrieval records in the last 14 days. If you do not want to restart Kafka Manager, wait for 14 days until the consumer group is automatically cleared.

8 Monitoring & Alarm

8.1 Why Can't I View the Monitoring Data?

If topic monitoring data is not displayed, the possible causes are as follows:

- The topic name starts with a special character, such as a number sign (#).
- No topic is created in the Kafka instance.

Solution:

- Delete topics whose names contain special characters.
- Create a topic.

If consumer group monitoring data is not displayed, the possible causes are as follows:

- The consumer group name starts with a special character, such as a number sign (#).
- No consumers in the group have connected to the instance.

Solution:

- Delete consumer groups whose names contain special characters.
- Consume messages using this consumer group.

8.2 Why Is the Monitored Number of Accumulated Messages Inconsistent with the Message Quantity Displayed on the Kafka Console?

Symptom: The monitoring data shows that there are 810 million accumulated messages. However, the Kafka console shows that there are 100 million messages in all six topics of the instance.

Analysis: The two statistics methods are different. The Kafka console shows the number of messages that have not been retrieved. The monitoring data shows the

number of accumulated messages in the topics multiplied by the number of consumer groups.

8.3 Why Is a Consumer Group Still on the Monitoring Page After Being Deleted?

The monitoring data is reported every minute. The reported data will be displayed on the monitoring page after being sorted. This process takes less than 20 minutes. After deleting a consumer group, you can wait for a while before checking the latest monitoring data.

8.4 Why Do Metrics Fluctuate Significantly (Disk Read/ Write Speed, Average Disk Read/Write Time, and CPU Usage)?

Metrics including **Disk Read Speed**, **Disk Write Speed**, **Average Disk Read Time**, **Average Disk Write Time**, and **CPU Usage** output instantaneous values. These metrics are for reference only. Generally, flushing Kafka data asynchronously to disk consumes disk I/O and CPU usage, causing fluctuations. They do not affect services.

8.5 Why Does JVM Heap Memory Usage of JVM Fluctuate Significantly?

It is normal that **JVM Heap Memory Usage of JVM** is displayed in a sawtooth shape on the monitoring page. The maximum value may further increase or the fluctuation may be greater after scale-out or an engine upgrade. Such fluctuations are caused by normal JVM heap memory reclamation, and do not affect services.

8.6 Why Is Production Rate Still 0 When There Are Produced Messages?

The metric data of the message production rate is reported once every minute. Message production rate = Number of messages produced per minute/60. The value is rounded down. The production rate is displayed as 0 if the number of messages produced within a minute is less than 60.